



# TP

## Test Plan

### Sapori di Unisa

Riferimento	NC1_TP
Versione	1.2
Data	01/02/2023
Destinatario	Prof. Carmine Gravino
Presentato da	Antonio Facchiano, Gianmarco Riviello, Salvatore Ruocco, Simone Vittoria
Approvato da	



## Revision History

Data	Versione	Descrizione	Autori
05/12/2023	0.1	Introduzione, Relazione con altri documenti, Panoramica del sistema, Materiale di testing, Approccio	Salvatore Ruocco
05/12/2023	0.3	Funzionalità da testare e non, Pass/Fail Criteria	Gianmarco Riviello
09/12/2023	0.5	Test Frame	Gianmarco Riviello Salvatore Ruocco
15/12/2023	1.0	Revisione e delibera documento	Gianmarco Riviello Salvatore Ruocco Simone Vittoria Antonio Facchiano
20/01/2024	1.1	Aggiunta di nuovi piani di test	Gianmarco Riviello, Salvatore Ruocco
01/02/2024	1.2	Revisione e delibera documento	Gianmarco Riviello Salvatore Ruocco Simone Vittoria Antonio Facchiano



## Sommario

1 Introduzione .....	4
1.1 Obiettivi .....	4
1.2 Definizioni, acronimi e abbreviazioni .....	4
1.3 Riferimenti .....	4
2 Relazione con gli altri documenti .....	5
3 Panoramica del sistema .....	5
4 Funzionalità da testare e non .....	6
5 Pass/Fail criteria .....	6
6 Approccio .....	7
6.1 Testing di sistema .....	7
6.2 Testing di integrazione .....	7
6.3 Testing di unità .....	7
7 Materiale di testing .....	8
8 Test Cases .....	8
9 Specifiche dei test cases .....	9
9.1 Gestione Magazzino .....	9
TC_1.1 Registra fornitura .....	9
9.2 Gestione Vendite .....	12
TC_2.1 Vendita Prodotti .....	12
9.3 Autenticazione .....	14
TC_3.1 Login .....	14
TC_3.2 Modifica Pin .....	14
9.4 Gestione Scaffale .....	15
TC_4.1 Aggiungi scaffale .....	15
9.5 Gestione Finanze .....	17
TC_5.1 Visualizza Bilancio .....	17
TC_5.2 Imposta sconto .....	18



# 1 Introduzione

---

## 1.1 Obiettivi

*Sapori di Unisa* è un supermercato che ha deciso di rinnovarsi digitalizzando la gestione della propria attività. I dirigenti hanno rilevato la necessità di modernizzare i processi; pertanto, hanno optato per l'introduzione di sistemi informatici per gestire le finanze e il magazzino. In particolare, hanno individuato delle discrepanze nella gestione dovute a errori umani, e per questo motivo hanno scelto di minimizzare tali errori adottando un software appositamente progettato.

Il sistema che si vuole realizzare ha come obiettivo principale quello di ottimizzare la gestione del magazzino e di avere un maggior controllo riguardo gli aspetti finanziari dell'attività. Attraverso la piattaforma, i dipendenti del supermercato potranno visionare in tempo reale la disponibilità dei prodotti presenti in magazzino e di conseguenza intraprendere le giuste azioni. Inoltre, il sistema consente di tener traccia degli aspetti finanziari e in particolare di controllare gli utili e le spese del supermercato.

Volendo rispettare questi obiettivi, non è possibile trascurare l'importanza di fornire al cliente un prodotto quanto più affidabile possibile. Sarà obiettivo di questo documento mettere in campo le strategie necessarie per verificare il comportamento di quello che sarà il sistema implementato rispetto a quanto descritto nei documenti.

Detto ciò, questo documento fa da apripista per l'intero team nelle attività di testing, dove sarà importante individuare failure e correggere appena possibile i relativi fault.

## 1.2 Definizioni, acronimi e abbreviazioni

Vengono riportati di seguito alcune definizioni presenti nel documento corrente:

- **SDD:** System Design Document
- **RAD:** Requirements Analysis Document
- **TP:** Test Plan

## 1.3 Riferimenti

- Libro di testo utilizzato durante il corso: [Object-Oriented Software Engineering Using UML, Patterns, and Java](#)
- [Statement Of Work](#)
- [RAD](#)
- [SDD](#)



## 2 Relazione con gli altri documenti

Documenti correlati al TP:

- **RAD**

La pianificazione dei casi di test non può non dipendere da quanto è stato descritto in fase di raccolta e analisi dei requisiti; quindi, si farà riferimento alle specifiche descritte nel RAD, in particolare ai requisiti funzionali, agli scenari e ai casi d'uso.

- **SDD**

Qui la nostra attenzione sarà rivolta verso la tabella che illustra i servizi dei sottosistemi per definire quali funzionalità saranno argomento di testing.

## 3 Panoramica del sistema

L'architettura del sistema è **three-tier**. Uno dei motivi che hanno influenzato tale scelta è sicuramente il background degli sviluppatori, i quali hanno già avuto a che fare con tale architettura. Inoltre, essa è molto indicata per lo sviluppo di applicazioni web, poiché si ha una netta separazione tra interfacce utente, logica applicativa e persistenza dei dati. Queste caratteristiche garantiscono una serie di qualità essenziali per lo sviluppo software odierno, su tutte la manutenzione e il riuso.

Per lo sviluppo del sistema verranno utilizzate le seguenti tecnologie.

Back-end	Front-end
<ul style="list-style-type: none"><li>• Java 21</li><li>• MySQL 5.6</li><li>• Apache Tomcat 10.1</li></ul>	<ul style="list-style-type: none"><li>• HTML5</li><li>• JSP 3.1</li><li>• JSTL 1.2.7</li><li>• CSS3</li><li>• JavaScript</li><li>• JQuery 3.7.1</li></ul>

Per la gestione delle librerie verrà usato **Maven**.



## 4 Funzionalità da testare e non

---

Di seguito si elencheranno le funzionalità alle quali verrà applicato il testing per i vari sottosistemi:

- Gestione Magazzino:
  - Registra fornitura
- Gestione Vendite:
  - Vendita prodotti
- Autenticazione:
  - Login
  - Modifica pin
- Gestione Scaffali:
  - Aggiungi prodotto scaffale
- Gestione Finanze:
  - Visualizza bilancio
  - Imposta sconto

Le funzionalità di cui non si andrà ad effettuare attività di testing riguardano requisiti funzionali di bassa o media priorità; sono inoltre escluse le funzionalità che non prevedono input manuale da parte dell'utente - ad esempio attività riguardanti esclusivamente la visualizzazione di dati o semplici click di un bottone.

## 5 Pass/Fail criteria

---

Il testing viene eseguito per rilevare eventuali anomalie del sistema proposto, in gergo tecnico **faults**, per poi andarli a correggere con specifiche modalità d'approccio e strumenti.

Il risultato atteso dei vari test case è confrontato con un oracolo (specifica del risultato atteso secondo i criteri progettuali, vincoli e requisiti funzionali richiesti).

In caso di successo, ovvero la situazione in cui c'è corrispondenza tra oracolo e test case, si ha il successo del test, **Pass**.

Nel caso contrario si ha un **Fail**.

Per cercare di minimizzare i difetti del software, si sa che c'è bisogno di una continua e precisa attività di verifica e convalida che va effettuata e applicata durante tutta la progettazione; per questo, il test ha dei requisiti ben precisi per essere ritenuto valido:

- Testare tutti i requisiti funzionali elencati nel punto 4;
- Eseguire test di regressione nel momento in cui si vanno ad aggiungere altre caratteristiche al sistema oppure modificando quelle già comprese in precedenza.



## 6 Approccio

---

Il testing dell'intero sistema prevede tre fasi:

- testing di sistema
- testing di integrazione
- testing di unità

Prima della fase di implementazione, avverrà la progettazione dei casi di test, perfezionati in seguito durante la loro fase di esecuzione.

### 6.1 Testing di sistema

Questa fase ha come obiettivo la definizione di casi di test che verificano l'affidabilità del sistema osservando il suo comportamento rispetto a ciò che l'utente si aspetta; in questa fase, non è di primaria importanza avere a disposizione il codice sorgente, infatti, si prevede l'utilizzo di una metodologia black-box. Nello specifico, si è deciso di adottare per il testing di sistema la metodologia Category Partition.

### 6.2 Testing di integrazione

Per il testing di integrazione si è deciso di procedere integrando le componenti, simulando quest'ultime attraverso dei mock (fantocci).

Per l'integrazione delle componenti è stata scelta la strategia Bottom-up, approccio ritenuto il più adatto per un software basato su sistemi real-time. Il testing di integrazione sarà lo stesso per tutte le componenti da testare.

### 6.3 Testing di unità

Per il testing di unità, a seguito dei vincoli previsti, il team testerà diversi metodi di alcune classi del sistema avendo a disposizione la piena conoscenza del codice implementato.



## 7 Materiale di testing

---

Come già anticipato, le attività di testing richiederanno il supporto dei documenti di progetto RAD e SDD in modo da individuare le componenti da testare.

L'esecuzione dei test dovrà avvenire su un server in locale, nel quale verrà eseguita l'applicazione. Su questo server dovranno essere configurate le tecnologie usate per lo sviluppo del sistema.

Nella fase di testing di sistema, verrà usato il tool Selenium, il quale automatizzerà le interazioni utente con le interfacce del sistema.

Invece, il testing di unità dovrà essere svolto tramite il tool JUnit e il testing di integrazione tramite il framework Mockito.

Per il monitoraggio dei Branch Coverage è stato scelto il tool JaCoCo.

## 8 Test Cases

---

Per sviluppare i test cases sarà utilizzato il metodo del Category Partition. Questo metodo consiste nell'identificare, per ogni funzionalità da testare, dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Alle scelte verrà assegnato un valore.

Quindi, per ogni requisito funzionale, dobbiamo:

1. Individuare una serie di parametri (input)
2. Per ogni parametro, individuare le categorie
3. Per ogni categoria, individuare le scelte
4. Andare a costruire i Test Frame: combinazioni di scelte (sensate) da ogni categoria individuata
5. Per ogni Test Frame, indicare l'esito: *errato* se non consente il raggiungimento dell'obiettivo del requisito, altrimenti *corretto*
6. Per ogni Test Frame, si crea un Test Case, che assegna dei valori precisi alle scelte

I test cases verranno definiti nel documento di Test Cases Specification (TCS).





## 9 Specifiche dei test cases

### 9.1 Gestione Magazzino

#### TC\_1.1 Registra fornitura

**Parametro:** Nome prodotto  
**Formato:** `^[a-zA-Z0-9\s]{2,255}$`

Categorie	Scelte
<b>Lunghezza [LN]</b>	<b>1:</b> Lunghezza <= 1 [errore] <b>2:</b> Lunghezza >= 2 && Lunghezza <= 255 [property <b>LunghezzaLN_OK</b> ] <b>3:</b> Lunghezza >= 256 [errore]
<b>Formato [FN]</b>	<b>1:</b> rispetta il formato [property <b>FormatoFN_OK</b> ] <b>2:</b> non rispetta il formato [errore]

**Parametro:** Marchio  
**Formato:** `^[a-zA-Z0-9\s]{2,255}$`

Categorie	Scelte
<b>Lunghezza [LM]</b>	<b>1:</b> Lunghezza <= 1 [errore] <b>2:</b> Lunghezza >= 2 && Lunghezza <= 255 [property <b>LunghezzaLM_OK</b> ] <b>3:</b> Lunghezza >= 256 [errore]
<b>Formato [FM]</b>	<b>1:</b> rispetta il formato [property <b>FormatoFM_OK</b> ] <b>2:</b> non rispetta il formato [errore]

**Parametro:** Prezzo di acquisto  
**Formato:** `^(\\d{1,5})(\\.\\d{0,2})?$`

Categorie	Scelte
<b>Valore [VA]</b>	<b>1:</b> Valore < 0.00 [errore] <b>2:</b> Valore >= 0.00 && Valore <= 99999.99 [property <b>ValoreVA_OK</b> ] <b>3:</b> Valore >= 100000.00 [errore]
<b>Formato [FA]</b>	<b>1:</b> rispetta il formato [property <b>FormatoFA_OK</b> ] <b>2:</b> non rispetta il formato [errore]



**Parametro:** Quantità

**Formato:** ^\d+\$

Categorie	Scelte
<b>Formato [FQ]</b>	<b>1:</b> rispetta il formato [property <b>FormatoFQ_OK</b> ] <b>2:</b> non rispetta il formato [errore]
<b>Valore [VQ]</b>	<b>1:</b> Valore <= 0 [errore] <b>2:</b> Valore >= 1000000 [errore] <b>3:</b> Valore >= 1 && Valore <= 999999 [property <b>Valore_VQ_OK</b> ]

**Parametro:** Data di scadenza

Categorie	Scelte
<b>Valida [VD]</b>	<b>1:</b> valida [property <b>ValidaVD_OK</b> ] <b>2:</b> non valida [errore]
<b>Correttezza [CD]</b>	<b>1:</b> dataDiScadenza < dataCorrente[errore] <b>2:</b> dataDiScadenza >= dataCorrente[property <b>CorrettezzaCD_OK</b> ]

**Parametro:** Foto

Categorie	Scelte
<b>Estensione [EI]</b>	<b>1:</b> estensione = PNG [property <b>EstensioneEI_OK</b> ] <b>2:</b> estensione != PNG [errore]
<b>Dimensioni [DI]</b>	<b>1:</b> dimensioni >= 2 MB [errore] <b>2:</b> dimensioni < 2 MB [property <b>DimensioniDI_OK</b> ]

Codice	Combinazione	Esito
TC_1.1.1	LN1	Errore: nome prodotto troppo corto
TC_1.1.2	LN3	Errore: nome prodotto troppo lungo
TC_1.1.3	LN2.FN2	Errore: nome prodotto formato non rispettato



TC_1.1.4	LN2.FN1.LM1	Errore: marchio troppo corto
TC_1.1.5	LN2.FN1.LM3	Errore: marchio troppo lungo
TC_1.1.6	LN2.FN1.LM2.FM2	Errore: marchio non rispettato
TC_1.1.7	LN2.FN1.LM2.FM1.VA1	Errore: prezzo acquisto insufficiente
TC_1.1.8	LN2.FN1.LM2.FM1.VA3	Errore: prezzo acquisto eccedente
TC_1.1.9	LN2.FN1.LM2.FM1.VA2.FA2	Errore: prezzo acquisto formato non valido
TC_1.1.10	LN2.FN1.LM2.FM1.VA1.FA1.FQ2	Errore: quantità non è un numero
TC_1.1.11	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ1	Errore: quantità minore di 0
TC_1.1.12	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ2	Errore: quantità eccedente
TC_1.1.13	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ3.VD2	Errore data scadenza formato non valido
TC_1.1.14	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ3.VD1.CD1	Errore: data scadenza precedente a quella attuale
TC_1.1.15	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ3.VD1.CD2.EI2	Errore: estensione immagine diversa da PNG
TC_1.1.16	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ3.VD1.CD2.EI1.DI1	Errore: dimensioni immagine superiore a 2 MB
TC_1.1.17	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ3.VD1.CD2.EI1.DI2	Corretto



## 9.2 Gestione Vendite

### TC\_2.1 Vendita Prodotti

Parametro: ID Prodotto	
Categorie	Scelte
Valido [VDID]	1: valida [property <b>ValidaVDID_OK</b> ] 2: non valida [errore]

Parametro: Quantità Formato: $^{\wedge}\backslash d+\$$	
Categorie	Scelte
Formato [FQ]	1: rispetta il formato [property <b>FormatoFQ_OK</b> ] 2: non rispetta il formato [errore]
Valore [VQ]	1: Valore $\leq 0$ [errore] 2: Valore $\geq 1000000$ [errore] 3: Valore $\geq 1$ && Valore $\leq 999999$ [property <b>ValoreVQ_OK</b> ]
QuantitàScaffali [QS]	1: quantitàScaffali – ValoreVQ $\geq 0$ [property <b>QuantitàScaffaliQS_OK</b> ] 2: quantitàScaffali – ValoreVQ $< 0$ [errore]

Parametro: Prezzo Formato: $^{\wedge}(\backslash d\{1,5\})(\backslash .\backslash d\{0,2\})?\$$	
Categorie	Scelte
Formato [FP]	1: rispetta il formato [property <b>FormatoFP_OK</b> ] 2: non rispetta il formato [errore]
Valore [VP]	1: Valore $\leq 0$ [errore] 2: Valore $\geq 100000.00$ [errore] 3: Valore $\geq 1$ && Valore $\leq 99999.99$ [property <b>ValoreVP_OK</b> ]

Codice	Combinazione	Esito
TC_2.1.1	VDID2	Errore nella selezione del prodotto.
TC_2.1.2	VDID1.FQ2	Errore: formato inserito errato.
TC_2.1.3	VDID1.FQ1.VQ1	Errore: la quantità inserita è sotto il limite minimo consentito.



TC_2.1.4	VDID1.FQ1.VQ2	Errore: la quantità inserita è sopra il massimo limite consentito.
TC_2.1.5	VDID1.FQ1.VQ3.QS2	Errore: la quantità selezionata del prodotto non è disponibile.
TC_2.1.6	VDID1.FQ1.VQ3.QS1.FP2	Errore: Il formato del prezzo non è giusto
TC_2.1.7	VDID1.FQ1.VQ3.QS1.FP1.VP1	Errore: Il prezzo è sotto la soglia minima consentita.
TC_2.1.8	VDID1.FQ1.VQ3.QS1.FP1.VP2	Errore: Il prezzo è sopra la soglia massima consentita.
TC_2.1.9	VDID1.FQ1.VQ3.QS1.FP1.VP3	Corretto.



## 9.3 Autenticazione

### TC\_3.1 Login

Parametro: Pin Formato: <code>^[0-9]{4}\$</code>	
Categorie	Scelte
Formato [FQ]	1: rispetta il formato [property <b>FormatoFQ_OK</b> ] 2: non rispetta il formato [errore]
Match [MC]	1: Match con pin = false [errore] 2: Match con pin = true [property <b>MatchMQ_OK</b> ]

Codice	Combinazione	Esito
TC_3.1.1	FQ2	Errore: Formato pin non rispettato
TC_3.1.2	FQ1.MC1	Errore: Nessun account corrisponde a questo pin
TC_3.1.3	FQ1.MC2	Corretto

### TC\_3.2 Modifica Pin

Parametro: Pin Formato: <code>^[0-9]{4}\$</code>	
Categorie	Scelte
Formato [FQ]	1: rispetta il formato [property <b>FormatoFQ_OK</b> ] 2: non rispetta il formato [errore]
Match [MC]	1: Match con pin attivi in quel momento = true[errore] 2: Match con pin != true [property <b>MatchMC_OK</b> ]

Codice	Combinazione	Esito
TC_3.2.1	FQ2	Errore: Formato pin non rispettato
TC_3.2.2	FQ1.MC1	Errore: Un account fra quelli esistenti ha già impostato questo pin, deve essere diverso.
TC_3.2.3	FQ1.MC2	Corretto



## 9.4 Gestione Scaffale

### TC\_4.1 Aggiungi scaffale

Parametro: Quantità Prodotto già Esposto Formato: ^\d+\$	
Categorie	Scelte
Valore [VQ]	1: Valore <= 0 [errore] 2: Valore >= 1000000 [errore] 3: Valore >= 1 && Valore <= 999999 [property <b>ValoreVQ_OK</b> ]
Formato [FQ]	1: rispetta il formato [property <b>FormatoFQ_OK</b> ] 2: non rispetta il formato [errore]
QuantitàMagazzino [QM]	1: quantità magazzino – ValoreVQ >= 0 [property <b>QuantitàMagazzinoQM_OK</b> ] 2: quantità magazzino – ValoreVQ < 0 [errore]

Parametro: Quantità Prodotto non esposto Formato: ^\d+\$	
Categorie	Scelte
Valore [VNE]	1: Valore <= 0 [errore] 2: Valore >= 1000000 [errore] 3: Valore >= 1 && Valore <= 999999 [property <b>ValoreVNE_OK</b> ]
Formato [FNE]	1: rispetta il formato [property <b>FormatoFNE_OK</b> ] 2: non rispetta il formato [errore]
QuantitàMagazzino [QNE]	1: quantità magazzino – ValoreVNE >= 0 [property <b>QuantitàMagazzinoQNE_OK</b> ] 2: quantità magazzino – ValoreVNE < 0 [errore]

Codice	Combinazione	Esito
TC_4.1.1	VQ1	Errore: valore quantità prodotto esposto minore di 0
TC_4.1.2	VQ2	Errore: valore quantità prodotto esposto è maggiore della massima consentita
TC_4.1.3	VQ3.FQ2	Errore: formato quantità esposti non valido



TC_4.1.4	VQ3.FQ1.QM2	Errore: quantità prodotto esposto è maggiore della quantità presente in magazzino
TC_4.1.5	VQ3.FQ1.QM1.VNE1	Errore: quantità del prodotto non esposto sotto il limite consentito
TC_4.1.6	VQ3.FQ1.QM1.VNE2	Errore: quantità del prodotto non esposto è maggiore della massima consentita
TC_4.1.7	VQ3.FQ1.QM1.VNE3.FNE2	Errore: formato quantità prodotti non esposti non corretto.
TC_4.1.8	VQ3.FQ1.QM1.VNE3.FNE1.QNE2	Errore: quantità prodotto non esposto insufficiente in magazzino
TC_4.1.9	VQ3.FQ1.QM1.VNE3.FNE1.QNE1	Corretto





## 9.5 Gestione Finanze

### TC\_5.1 Visualizza Bilancio

**Parametro: Data inizio**

Categorie	Scelte
<b>Valida [VI]</b>	<b>1:</b> valida [property <b>ValidaVI_OK</b> ] <b>2:</b> non valida [errore]
<b>Correttezza [CD]</b>	<b>1:</b> dataInizio >= data_oggi [errore] <b>2:</b> dataInizio < data_oggi [property <b>CorrettezzaCD_OK</b> ]

**Parametro: Data fine**

Categorie	Scelte
<b>Valida [VF]</b>	<b>1:</b> valida [property <b>ValidaVF_OK</b> ] <b>2:</b> non valida [errore]
<b>Correttezza [CF]</b>	<b>1:</b> dataFine < dataInizio [errore] <b>2:</b> dataFine >= dataInizio && dataFine < data_oggi [property <b>CorrettezzaCF_OK</b> ] <b>3:</b> dataFine >= dataInizio && dataFine >= data_oggi [errore]

Codice	Combinazione	Esito
TC_5.1.1	VI2	Errore: data inizio non valida
TC_5.1.2	VI1.CD1	Errore: data inizio non può essere uguale o maggiore della data odierna.
TC_5.1.3	VI1.CD2.VF2	Errore: data fine non valida
TC_5.1.4	VI1.CD2.VF1.CF1	Errore: data fine precede quella di inizio
TC_5.1.5	VI1.CD2.VF1.CF3	Errore: data fine non può essere uguale o maggiore a quella odierna
TC_5.1.6	VI1.CD2.VF1.CF2.	Corretto.



## TC\_5.2 Imposta sconto

### Parametro: ID prodotto

Categorie	Scelte
<b>Valido [VP]</b>	1: valido [property <b>ValidoVP_OK</b> ] 2: non valido [errore]
<b>NotGiàInSconto[NGS]</b>	1: dataFineScontoProdotto == null    dataFineScontoProdotto < dataInizio [property <b>NotGiàInScontoNGS_OK</b> ] 2: dataFineScontoProdotto >= dataInizio [errore]

### Parametro: Data inizio

Categorie	Scelte
<b>Valida [VI]</b>	1: valida [property <b>ValidaVI_OK</b> ] 2: non valida [errore]
<b>Futuro [FI]</b>	1: dataInizio > dataCorrente [property <b>CorrettezzaFI_OK</b> ] 2: dataInizio <= dataCorrente[errore]

### Parametro: Data fine

Categorie	Scelte
<b>Valida [VF]</b>	1: valida [property <b>ValidaVF_OK</b> ] 2: non valida [errore]
<b>Correttezza [CF]</b>	1: dataFine > dataInizio [property <b>CorrettezzaCF_OK</b> ] 2: dataFine <= dataInizio [errore]

### Parametro: Percentuale Sconto

Formato: ^\d+\$

Categorie	Scelte
<b>Valore [VA]</b>	1: Valore < 0 [errore] 2: Valore >= 1 && Valore <= 100 [property <b>ValoreVA_OK</b> ] 3: Valore > 100 [errore]
<b>Formato [FP]</b>	1: rispetta il formato [property <b>FormatoFP_OK</b> ] 2: non rispetta il formato [errore]



Codice	Combinazione	Esito
TC_5.2.1	VP2	Errore: id prodotto non valido
TC_5.2.2	VP1.NGS2	Errore: prodotto già in sconto
TC_5.2.3	VP1.NGS1.VI2	Errore: data inizio non valida
TC_5.2.4	VP1.NGS1.VI1.FI2	Errore: data inizio antecedente a quella odierna
TC_5.2.5	VP1.NGS1.VI1.FI1.VF2	Errore: data fine non valida
TC_5.2.6	VP1.NGS1.VI1.FI1.VF1.CF2	Errore: data fine antecedente a quella di inizio
TC_5.2.7	VP1.NGS1.VI1.FI1.VF1.CF1.VA1	Errore: valore della percentuale sconto troppo bassa
TC_5.2.8	VP1.NGS1.VI1.FI1.VF1.CF1.VA3	Errore: valore della percentuale sconto troppo alto
TC_5.2.9	VP1.NGS1.VI1.FI1.VF1.CF1.VA2.FP2	Errore: formato della percentuale sconto non rispettato
TC_5.2.10	VP1.NGS1.VI1.FI1.VF1.CF1.VA2.FP1	Corretto